10/500197

PCT JP02/13442

10 Rec'd PCT/PTO 24 JUN 2004

日本国特許庁
JAPAN PATENT OFFICE

28.02.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年１２月２８日

出願番号
Application Number: 特願２００１－４０１４６２

ST.10/C]: [ＪＰ２００１－４０１４６２]

出願人
Applicant(s): 東京エレクトロンデバイス株式会社
西原　明法

２００３年　１月１７日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００２－３１０７１３９

【書類名】　特許願
【整理番号】　JPP014008
【提出日】　平成13年12月28日
【あて先】　特許庁長官　殿
【国際特許分類】　G06F　9/00
【発明者】
　【住所又は居所】　神奈川県横浜市都筑区東方町１番地　東京エレクトロンデバイス株式会社内
　【氏名】　三田　高司
【特許出願人】
　【識別番号】　500323188
　【氏名又は名称】　東京エレクトロンデバイス株式会社
【特許出願人】
　【識別番号】　501186977
　【氏名又は名称】　西原　明法
【代理人】
　【識別番号】　100095407
　【弁理士】
　【氏名又は名称】　木村　満
【手数料の表示】
　【予納台帳番号】　038380
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　0014440
【プルーフの要否】　要

【書類名】 明細書

【発明の名称】 演算システム

【特許請求の範囲】

【請求項１】

ルックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、

各前記プログラム記憶手段のいずれかを選択する選択手段と、

複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の１以上に入力することで、該プログラムモジュールに応じた演算を実行する論理演算手段と、

を備えることを特徴とする演算システム。

【請求項２】

前記選択手段は、シフトレジスタより構成されている、

ことを特徴とする請求項１に記載の演算システム。

【請求項３】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させる、

ことを特徴とする請求項１又は２に記載の演算システム。

【請求項４】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備える、

ことを特徴とする請求項１、２又は３に記載の演算システム。

【請求項５】

前記論理演算手段の内部状態を退避する退避手段と、

所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、

を更に備えることを特徴とする請求項４に記載の演算システム。

【請求項６】

前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含み、

前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備え、

前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させる

ことを特徴とする請求項５に記載の演算システム。

【請求項７】

前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備える

ことを特徴とする請求項５又は６に記載の演算システム。

【請求項８】

前記退避手段は、先入れ後出し方式のスタックによって構成される

ことを特徴とする請求項５、６又は７に記載の演算システム。

【請求項９】

前記プログラム記憶手段に記憶された各プログラムモジュールは、前記論理演算手段を構成する論理回路に入力する信号に応じたコードによって構成されている

ことを特徴とする請求項１乃至８のいずれか１項に記載の演算システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、プログラムの実行をハードウェアで直接的に実現できる演算システムに関し、特に大規模プログラムの実行に適した演算システムに関する。

【０００２】

【従来の技術】

現在の汎用コンピュータは、ＣＰＵ（Central Processing Unit）がメモリに記憶されたプログラム中の命令を順次解釈しながら、演算を進めていく。ＣＰＵは、プログラムで実行すべき演算をソフトウェアで実現するものであり、必ずしもその演算に対して最適なハードウェア構成となっていないため、最終的な演算結果を得るまでに多くのオーバーヘッドが存在する。

【０００３】

これに対して、プログラムの実行をハードウェアで直接的に実現するための技術として、例えば、特表平８－５０４２８５号公報（国際公開ＷＯ９４／１０６２７号公報）や特表２０００－５１６４１８号公報（国際公開ＷＯ９８／０８３０６号公報）に示されているような、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）を利用した演算システムが知られている。

【０００４】

ＦＰＧＡは、プログラムとして論理データを与えることで論理回路間の結線論理を変更し、これによってハードウェア的に演算結果を得ることをできるようにしたものである。ＦＰＧＡを利用して演算を行うことによって、特定の演算専用に構成されたハードウェア回路ほどは高速ではないが、従来の汎用コンピュータ

のようなＣＰＵによる演算に比べると、非常に高速に演算結果を得ることができる。

【０００５】

【発明が解決しようとする課題】

ところで、現在の汎用コンピュータで実行されているプログラム、特に大規模なプログラムは、複数のモジュールに分割して作成されている。そして、１のプログラムモジュールが他のプログラムモジュールを呼び出しながら、全体としてのプログラムの実行を進めていくようになっている。こうしてプログラムモジュール別に開発を進めたり、各プログラムモジュールを部品として利用したりすることにより、プログラムの開発期間を短縮することができる。

【０００６】

しかしながら、上記した従来のＦＰＧＡを用いた演算システムでは、ハードウェアとしてのモジュール分割は考えられていても、ソフトウェアとしてのモジュール分割は考えられていなかった。つまり、ソフトウェアとして１のプログラムモジュールから他のプログラムモジュールを呼び出し、呼び出したプログラムモジュールの実行を終了した後、元のプログラムモジュールに復帰するというように、複数のプログラムモジュールを適時実行していくことにより大規模プログラムの実行を可能とする仕組みは考えられていなかった。

【０００７】

また、ソフトウェアとしてのモジュール分割を行うとしても、プログラムモジュールを呼び出すごとにＦＰＧＡのコンフィギュレーションを逐次書き換えるようにした場合、書き換えの時間が長いため、オーバーヘッドの多い非効率的な演算システムとなってしまう。

【０００８】

このため、従来のＦＰＧＡを用いた演算システムで実行可能なプログラムは、実質的に１のみのモジュールで作成されたプログラムでなくてはならないという制約があった。つまり、大規模プログラムの実行が事実上不可能で、その適用範囲は限られるという問題があった。

【０００９】

本発明は、上記した従来技術の問題点を解消するためになされたものであり、汎用のＣＰＵを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現した演算システムを提供することを目的とする。

【００１０】

【課題を解決するための手段】

上記目的を達成するため、本発明の演算システムは、

ルックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、

各前記プログラム記憶手段のいずれかを選択する選択手段と、

複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の１以上に入力することで、該プログラムモジュールに応じた演算を実行する論理演算手段と、

を備えることを特徴とする。

【００１１】

上記演算システムでは論理演算手段のコンフィギュレーションの書き換えによるオーバーヘッドがほとんど生じないため、演算が効率的になり、従って高速になる。

【００１２】

なお、前記選択手段は、たとえば、シフトレジスタより構成されていればよい。

また、各前記プログラム記憶手段が、プログラムモジュールを各々書き換え可能に記憶するものである場合、前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させるものであってもよい。

【００１３】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記

憶するものであってもよい。

この場合、前記演算システムは、前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備えるものとすれば、プログラムモジュールのデータ量の総量が、プログラム記憶手段の記憶容量より大きくても、このプログラムモジュールをすべて利用することが可能になる。

【００１４】

前記演算システムは、

前記論理演算手段の内部状態を退避する退避手段と、

所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、

を更に備えていてもよい。

このような構成を有する演算システムは、メモリにロードするプログラムモジュールを切り替えるときに、論理演算手段の内部状態を退避し、また、復元する仕組みを備えている。このため、複数のプログラムモジュールからなる大規模なプログラムも、メモリにロードするプログラムを切り替え、論理演算手段中の論理回路に入力する信号を変えていくことで、ハードウェア的に高速に演算を実行していくことができる。

【００１５】

上記演算システムにおいて、

前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含むものであってもよい。

この場合において、上記演算システムは、

前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備えるものとすることができ、

前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させるものとすることができる。

【００１６】

ここで、前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備えていてもよい。

【００１７】

これらの仕組みをさらに備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に高速に実行することが可能となる。

【００１８】

上記演算システムにおいて、

前記退避手段は、先入れ後出し方式のスタックによって構成されたものとすることができる。

【００１９】

このようなスタックで構成される退避手段により、例えば、別のプログラムモジュールから呼び出されたプログラムモジュールが、さらに他のプログラムモジュールを呼び出すといった処理も可能となる。また、あるプログラムモジュールが、そのプログラムモジュール自身を呼び出す再帰型のプログラムを実行することも可能となる。

【００２０】

上記演算システムにおいて、

前記プログラム記憶手段に記憶された各プログラムモジュールは、具体的には、たとえば、前記論理演算手段を構成する論理回路に入力する信号に応じたコー

ドによって構成されている。

【００２１】

なお、各プログラムモジュール中の命令を構成するコードは、ハードウェア記述が可能な言語で記述されたソースプログラムをコンパイルすることによって得ることができる。この場合、モジュール別にソースプログラムを開発したり、モジュールの部品としての利用が可能となり、プログラムの開発期間を短縮することが可能となる。

【００２２】

【発明の実施の形態】

以下、添付図面を参照して、本発明の実施の形態について説明する。

【００２３】

図１は、この実施の形態にかかる演算システムの構成を示すブロック図である。図示するように、この演算システム１は、ＦＰＧＡデータ記憶部２と、ローダ３と、ＦＰＧＡデバイス４とから構成されている。ＦＰＧＡデータ記憶部２には、複数のモジュールに分かれたＦＰＧＡデータモジュール２－１～２－ｎを記憶している。

【００２４】

ＦＰＧＡデータモジュール２－１～２－ｎは、それぞれハードウェア記述が可能なプログラム言語で記述されている複数のモジュールに分かれたソースプログラム５－１～５－ｎを、ＦＰＧＡデバイス４の論理記述を行うべくコンパイラ６がコンパイルしたモジュール毎のデータである。ＦＰＧＡデータモジュール２－１～２－ｎはそれぞれ、入力データの値と出力データの値との対応関係を示すルックアップテーブルの形式をとるデータを含んでいる。

ソースプログラム５－１～５－ｎのうちの少なくとも１のモジュールは、他のモジュールのソースプログラム５－１～５－ｎを呼び出す機能を含んでおり、ＦＰＧＡデータモジュール２－１～２－ｎには、他のモジュールの呼び出しのためのデータも含まれている。

【００２５】

ローダ３は、論理回路等より構成されており、ＦＰＧＡデータ記憶部２に記憶

されたＦＰＧＡデータモジュール２－１～２－ｎをモジュール単位でＦＰＧＡデバイス４にロードする。

ローダ３によるＦＰＧＡデータモジュール２－１～２－ｎのロードの指示は、演算の実行の開始時に外部から与えられる他、ＦＰＧＡデバイス４による演算の実行によっても与えられる。ローダ３は、ロードの指示１回につき、この指示に従って、後述のＦＰＧＡデータメモリ４１ａ～４１ｄのいずれかに、ＦＰＧＡデータモジュールをロードする。

【００２６】

ＦＰＧＡデバイス４は、ローダ３によってロードされたＦＰＧＡデータモジュール２－１～２－ｎに従って論理構成を行い、外部からの入力データに所定の演算を施して出力データとして出力するもので、シフトレジスタ４０と、ゲートアレイ４３と、呼び出し検出部４４と、退避スタック４５と、引数受け渡し部４６と、制御部４７とを備えている。呼び出し検出部４４、退避スタック４５、引数受け渡し部４６及び制御部４７は、論理回路等より構成されている。

【００２７】

シフトレジスタ４０は、４個のＦＰＧＡデータメモリ４１ａ～４１ｄを含んでいる。ＦＰＧＡデータメモリ４１ａ～４１ｄはそれぞれ半導体メモリあるいはフリップフロップ回路などより構成されており、ローダ３がロードしたＦＰＧＡデータモジュールを記憶するための記憶領域を有している。

【００２８】

ＦＰＧＡデータメモリ４１ａ～４１ｄは、たとえば、それぞれデータバスと、アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備える。そして、各自のライトイネーブル端子にアクティブレベルの信号が供給されているとき、データバスに供給されたデータを、アドレスバスに供給されたアドレスが示す記憶領域に格納する。

なお、ローダ３は、ロードの指示を供給されるたびに、例えば、ＦＰＧＡデータモジュールをロードする記憶領域を示すアドレスをＦＰＧＡデータメモリ４１ａ～４１ｄのアドレスバスに供給し、また、当該ロードの指示が示す特定のＦＰＧＡデータメモリのライトイネーブル端子にアクティブレベルの信号を供給して

、この状態で、ＦＰＧＡデータメモリ４１ａ～４１ｄに、ロードすべきＦＰＧＡデータモジュールを供給するものとする。

【００２９】

また、シフトレジスタ４０は、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちデータを出力する対象であるＦＰＧＡデータメモリ１個を、制御部４７より与えられる指示に従って、サイクリックに指定する（すなわち、指定する対象を循環シフトさせる）。そして、指定したＦＰＧＡデータメモリが格納しているデータを、データバスを介して出力する。

ＦＰＧＡデータメモリ４１ａ～４１ｄが、上述したデータバス、アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してデータバスを介して出力する場合、シフトレジスタ４０は、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちいずれかのリードイネーブル端子にアクティブレベルの信号を供給するものとする。そして、指示を供給されるたびに、例えば、アクティブレベルの信号の供給先であるイネーブル端子を、計４個のリードイネーブル端子のうちからサイクリックに切り替えるものとする。なお、シフトレジスタ４０は、複数のＦＰＧＡデータメモリのリードイネーブル端子に同時にアクティブレベルの信号を供給して、当該複数のＦＰＧＡデータメモリから一括してデータを読み出すことが可能なように構成されていてもよい。

【００３０】

ゲートアレイ４３は、ＡＮＤ、ＯＲ、ＮＯＴなどの複数のゲート回路４３ａと、演算の途中結果を内部状態として保持している複数のフリップフロップ４３ｂとを含んでいる。各ゲート回路４３ａの出力論理は、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちシフトレジスタ４０により指定されたものに記憶されたＦＰＧＡデータモジュールに従って変更される。また、各フリップフロップ４３ｂは、所望のデータを外部から書き込むことができるようになっている。

【００３１】

呼び出し検出部４４は、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちセレクタ

４２により選択されたものに記憶されたＦＰＧＡデータモジュールに含まれる他のモジュールの呼び出しのためのデータを検出する。退避スタック４５は、呼び出し検出部４４によって他のモジュールの呼び出しのためのデータが検出されたとき、ゲートアレイ４３中のフリップフロップ４３ｂに保持されているデータと、呼び出し元のＦＰＧＡデータモジュールの識別データとを退避するためのスタックである。退避の手法としては、例えば、先入れ後出し方式を用いる。

【００３２】

引数受け渡し部４６は、モジュールの呼び出し、復帰の際において呼び出し元と呼び出し先のＦＰＧＡデータモジュール間における引数の受け渡しを行うものである。より詳細に説明すると、呼び出しの際には、呼び出し元のＦＰＧＡデータモジュールに従った演算の途中結果としてフリップフロップ４３ｂの所定のものに保持されていたデータを、呼び出し先のＦＰＧＡデータモジュールに従った演算の入力（引数）として与える。復帰の際には、呼び出し先のＦＰＧＡデータモジュールに従った演算結果（戻り値）の出力データを、ゲートアレイ４３中のフリップフロップ４３ｂの所定のものに書き込む。

【００３３】

制御部４７は、呼び出し検出部４４が他のモジュールの呼び出しのためのデータを検出した場合、当該呼び出しのためのデータの前までのＦＰＧＡデータモジュールに従った演算の途中結果としてフリップフロップ４３ｂのそれぞれに保持されているデータと、呼び出し元のデータモジュールの識別データとを退避スタック４５に退避させると共に、呼び出し先のＦＰＧＡデータモジュールに従った演算で使用するデータを保持するフリップフロップ４３ｂのデータを、引数受け渡し部４６に一時保持させる。

その後、ＦＰＧＡデータメモリ４１ａ～４１ｄのうち呼び出し先のＦＰＧＡデータモジュールを記憶するものがあれば、シフトレジスタ４０に、該当するＦＰＧＡデータメモリが指定されるまで指定する対象を繰り返し循環シフトさせ、なければ、ローダ３にロードさせる。そして、引数受け渡し部４６に一時保持したデータをゲートアレイ４３に入力データとして与える。

【００３４】

制御部４７は、また、呼び出されたＦＰＧＡデータモジュールに従った演算が終了したときに、その出力データを引数受け渡し部４６に一時保持させる。

その後、退避スタック４５に退避された呼び出し元のデータモジュールの識別データに従って、シフトレジスタ４０に循環シフトを行わせることにより呼び出し元のＦＰＧＡデータモジュールを指定させ、退避スタック４５に退避されたデータをフリップフロップ４３ｂに復帰させると共に、引数受け渡し部４６に一時保持させたデータをフリップフロップ４３ｂの所定のものに書き込ませる。

【００３５】

なお、ＦＰＧＡデバイス４に外部から入力される入力データは、キーボードなどの入力装置から入力されるデータの他、磁気ディスク装置などの外部記憶装置から読み出されたデータであってもよい。また、ＦＰＧＡデバイス４から外部に出力される出力データは、ディスプレイ装置などの出力装置から出力する他、外部記憶装置に書き込むものであってもよく、さらに、周辺機器を制御するための制御データであってもよい。

【００３６】

以下、この実施の形態にかかる演算システムにおける動作について、具体的な例に基づいて説明する。ここでは、ＦＰＧＡデータモジュール２－１が最初にロードされるものとし、ＦＰＧＡデータモジュール２－１は、ＦＰＧＡデータモジュール２－ｎを呼び出すものとする。また、退避スタック４５は、先入れ後出し方式でデータの退避を行うものとする。

【００３７】

ＦＰＧＡデータモジュール２－１がＦＰＧＡデータメモリ４１ａ～４１ｄのいずれかにロードされ（以下では、理解を容易にするため、ＦＰＧＡデータメモリ４１ａにロードされたものとする）、セレクタ４２がＦＰＧＡデータメモリ４１ａを選択すると、これに従ったレベルの信号がゲート回路４３ａに入力され、ゲートアレイ４３を構成するゲート回路４３ａが論理構成される。そして、ゲートアレイ４３に外部からの入力データが入力されることによって、ＦＰＧＡデータモジュール２－１に応じた演算がゲートアレイ４３において実行される。

【００３８】

一方、呼び出し検出部４４は、ＦＰＧＡデータメモリ４１ａにロードされたＦＰＧＡデータモジュール２－１にＦＰＧＡデータモジュール２－ｎを呼び出すためのデータが含まれていることを検出し、その旨を制御部４７に通知する。制御部４７は、その呼び出しにかかる部分の直前までの演算の途中結果としてフリップフロップ４３ｂに保持されているデータ（ゲートアレイ４３の内部状態）を、呼び出し元のＦＰＧＡデータモジュール２－１を識別するためのデータと共に退避スタック４５の一番上に退避させる。また、フリップフロップ４３ｂに保持されているデータのうちで呼び出し先のＦＰＧＡデータモジュール２－ｎに引数として渡すものを、引数受け渡し部４６に一時保存させる。

【００３９】

その後、制御部４７は、ＦＰＧＡデータモジュール２－ｎがＦＰＧＡデータメモリ４１ｂ～４１ｄのいずれかにロードされているか否かを判別する。（具体的には、たとえば、ＦＰＧＡデータモジュール２－ｎをロードする旨の指示を既に自らローダ３に供給したか否かを判別する。）そして、ロードされていなければ、ローダ３を制御し、呼び出し先であるＦＰＧＡデータモジュール２－ｎをＦＰＧＡデータメモリ４１ｂ～４１ｄのいずれかにロードさせ（以下では、理解を容易にするため、ＦＰＧＡデータメモリ４１ｂにロードされたものとする）、シフトレジスタ４０にＦＰＧＡデータメモリ４１ｂを指定させる。一方、ロードされていれば、シフトレジスタ４０に、ＦＰＧＡデータモジュール２－ｎがロードされているＦＰＧＡデータメモリ（以下では、理解を容易にするため、ＦＰＧＡデータメモリ４１ｂがこれに該当するものとする）を指定させる。

【００４０】

ＦＰＧＡデータモジュール２－ｎがＦＰＧＡデータメモリ４１ｂにロードされ、シフトレジスタ４０がＦＰＧＡデータメモリ４１ｂを指定すると、これに従ったレベルの信号がゲート回路４３ａに入力され、ゲートアレイ４３を構成するゲート回路４３ａが論理構成される。また、引数受け渡し部４６に引数として一時保存されたデータが、入力データとしてゲートアレイ４３に入力され、ＦＰＧＡデータモジュール２－ｎに応じた演算がゲートアレイ４３において実行される。

【００４１】

この演算が終了すると、制御部４７は、ゲートアレイ４３からの出力データを呼び出し元のＦＰＧＡデータモジュール２－１に渡す引数として引数受け渡し部４６に一時保存させる。制御部４７は、さらに退避スタック４５の一番上に退避されたデータを参照することでシフトレジスタ４０を制御して循環シフトをおこなわせ、呼び出し元のＦＰＧＡデータモジュール２－１を記憶するＦＰＧＡデータメモリ４１ａを再び指定させる。

【００４２】

呼び出し元のＦＰＧＡデータモジュール２－１がロードされたＦＰＧＡデータメモリ４１ａが再び指定されると、制御部４７は、退避スタック４５の一番上に退避されていた内部状態のデータをフリップフロップ４３ｂのそれぞれに書き戻し、ゲートアレイ４３の内部状態を復元させる。さらに、引数受け渡し部４６に引数として一時保存されていたデータをフリップフロップ４３ｂの所定のものに書き込む。この状態でゲートアレイ４３においてＦＰＧＡデータモジュール２－１に従った演算が再開され、最終的な演算結果が出力データとして出力されることとなる。

【００４３】

なお、ＦＰＧＡデータモジュール２－１から呼び出されたＦＰＧＡデータモジュール２－ｎが、さらに他のＦＰＧＡデータモジュールを呼び出すものであっても演算を実行することができる。ＦＰＧＡデータモジュール２－ｎがさらに他のモジュールを呼び出すことを呼び出し検出部４４が検出した場合にも、制御部４７は、上記と同じような制御を行うものとすればよい。

【００４４】

以上説明したように、この実施の形態にかかる演算システムでは、ゲートアレイ４３の内部状態（フリップフロップ４３ｂが保持するデータ）を退避スタック４５に退避した後に、実行中のモジュールとは異なるＦＰＧＡデータモジュールがＦＰＧＡデータメモリ４１ａ～４１ｄのいずれかにロードされ、ロードされたＦＰＧＡデータメモリの記憶内容が出力されるようにしている。また、退避スタック４５に退避した状態をゲートアレイ４３に復元してから元のモジュールに復帰することができるようになっている。

このため、各ＦＰＧＡデータモジュールをＦＰＧＡデータメモリに適宜ロードしていくことによって、複数のモジュールからなる大規模なプログラムを、各モジュールに対応してゲート回路４３ａ間の論理構成を変化させてハードウェア的に実行することができ、従来のＣＰＵを用いた演算システムに比べて高速に演算を実行することができる。

【００４５】

さらに、この演算システムはＦＰＧＡデータメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、ＦＰＧＡデータメモリが単一でＦＰＧＡデータモジュールの呼び出しがある毎に新たなＦＰＧＡデータモジュールがロードし直される構成に比べても演算が効率的で高速になる。

【００４６】

また、ＦＰＧＡデータモジュール２－１～２－ｎのうちの少なくとも１のモジュールが他のモジュールを呼び出すためのデータを含んでいるが、このような他のモジュールの呼び出しを含むＦＰＧＡデータモジュールがＦＰＧＡデータメモリにロードされた場合に、これを呼び出し検出部４４が検出している。そして、この検出結果に基づいて、退避スタック４５へのゲートアレイ４３の内部状態（フリップフロップ４３ｂが保持するデータ）の退避、引数受け渡し部４６を介した引数の受け渡しを行っている。また、呼び出し先のモジュールに従った演算が終了したときに、退避スタック４５に退避した内部状態の復元、引数受け渡し部４６を介した呼び出し元のモジュールへの引数の受け渡しを行っている。このような仕組みを備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に実行することが可能となる。

【００４７】

また、呼び出し検出部４４が他のモジュールの呼び出しを検出したときに、ゲートアレイ４３の内部状態（フリップフロップ４３ｂが保持するデータ）を退避するのは、先入れ後出し方式の退避スタックである。このため、他のモジュールから呼び出されたモジュールがさらに他のモジュールを呼び出すようなプログラムを実行することもできる。さらに、実行中のモジュールが自身を呼び出す再帰

型のプログラムを実行することもできる。

【００４８】

さらに、ＦＰＧＡデータモジュール２－１～２－ｎは、モジュール分割されたソースプログラム５－１～５－ｎをそれぞれコンパイラ６によってコンパイルしたものである。以上のような特徴を有することによって、この演算システムにおいて実行すべきプログラムは、モジュール別にソースプログラムの開発を進めたり、ソースプログラムの各モジュールを部品として利用したりすることが可能となり、その開発期間を短縮することができる。

【００４９】

本発明は、上記の実施の形態に限られず、種々の変形、応用が可能である。以下、本発明に適用可能な上記の実施の形態の変形態様について説明する。

【００５０】

上記の実施の形態では、ローダ３は、ＦＰＧＡデータ記憶部２に記憶されたいずれかのＦＰＧＡデータモジュール２－１～２－ｎを、そのままＦＰＧＡデータメモリ４１ａ～４１ｄにロードするものとしていた。これに対して、ＦＰＧＡデータモジュール２－１～２－ｎがマクロを含み、ＦＰＧＡデータ記憶部２にマクロデータを記憶させておき、ローダ３がＦＰＧＡデータメモリ４１ａ～４１ｄにロードする際に、マクロ展開をするものとしてもよい。

【００５１】

また、ＦＰＧＡデータメモリ４１ａ～４１ｄはシフトレジスタ４０とは別個のものとして、それぞれＲＡＭ（Random Access Memory）等のランダムアクセス可能なメモリより構成されていてもよい。

そしてこの場合、例えば図２に示すように、この演算システムは、シフトレジスタ４０に代えてセレクタ４２を備えるものとしてもよい。

【００５２】

この場合、ＦＰＧＡデータメモリ４１ａ～４１ｄは、たとえば、それぞれデータ／アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備えるものとする。そして、各自のライトイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスにまず供給されたアドレスが示す

記憶領域に、このデータ／アドレスバスに次に供給されたデータを格納する。また、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出し、このデータ／アドレスバスを介して出力するものとする。

ＦＰＧＡデータメモリ４１ａ～４１ｄがこのような構成を有している場合、ローダ３は、ロードの指示を供給されるたびに、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちこの指示が示すもののライトイネーブル端子にアクティブレベルの信号を供給し、この状態で、ＦＰＧＡデータメモリ４１ａ～４１ｄに、ロードすべきＦＰＧＡデータモジュールを供給すればよい。

【００５３】

セレクタ４２は、シフトレジスタ等より構成されており、ＦＰＧＡデバイス４による演算の実行によって制御部４７より与えられる指示に従い、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちこの指示が示すもの１個を指定する。（すなわち、ゲートアレイ４３や呼び出し検出部４４がデータを参照することが可能な状態とする。）

ＦＰＧＡデータメモリ４１ａ～４１ｄが、上述したデータ／アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してこのデータ／アドレスバスを介して出力する場合、セレクタ４２は、指示を供給されるたびに、ＦＰＧＡデータメモリ４１ａ～４１ｄのうちこの指示が示すもののリードイネーブル端子にアクティブレベルの信号を供給すればよい。

【００５４】

また、ＦＰＧＡデータメモリ４１ａ～４１ｄがデータを書き換え可能に記憶するメモリより構成されている場合、ＦＰＧＡデータメモリ４１ａ～４１ｄは、データの入力及び出力に共用されるデータバスに代えて、書き込む対象のデータを入力するための入力ポートと、読み出したデータを出力するための出力ポートとを、互いに別個のものとして備えるようにしてもよい。

そしてこの場合、ＦＰＧＡデータメモリ４１ａ～４１ｄは、ひとつのＦＰＧＡ

データメモリの出力ポートが他のＦＰＧＡデータメモリの入力ポートに接続されるようにして、全体として環をなすように接続されていてもよい。

具体的には、例えば図３に示すように、、ＦＰＧＡデータメモリ４１ａの出力ポートＯＵＴがＦＰＧＡデータメモリ４１ｂの入力ポートＩＮに接続され、ＦＰＧＡデータメモリ４１ｂの出力ポートＯＵＴがＦＰＧＡデータメモリ４１ｃの入力ポートＩＮに接続され、ＦＰＧＡデータメモリ４１ｃの出力ポートＯＵＴがＦＰＧＡデータメモリ４１ｄの入力ポートＩＮに接続され、ＦＰＧＡデータメモリ４１ｄの出力ポートＯＵＴがＦＰＧＡデータメモリ４１ａの入力ポートＩＮに接続されていてもよい。

【００５５】

そして、シフトレジスタ４０は、データを出力させる対象のＦＰＧＡデータメモリをサイクリックに指定する代わりに、ＦＰＧＡデータメモリ４１ａ～４１ｄに記憶されているデータ自体をＦＰＧＡデータメモリ４１ａ～４１ｄ相互間でサイクリックに転送させ、ゲートアレイ４３には、所定のＦＰＧＡデータメモリが記憶しているデータを供給するものとしてもよい。

具体的には、例えば図３に模式的に示すように、シフトレジスタ４０がデータバスを介して出力するデータは常にＦＰＧＡデータメモリ４１ａが記憶するデータであるものとし、シフトレジスタ４０は、制御部４７から指示が１回与えられる毎に、ＦＰＧＡデータメモリ４１ａが記憶していた内容をＦＰＧＡデータメモリ４１ｂに記憶させ、ＦＰＧＡデータメモリ４１ｂが記憶していた内容をＦＰＧＡデータメモリ４１ｃに記憶させ、ＦＰＧＡデータメモリ４１ｃが記憶していた内容をＦＰＧＡデータメモリ４１ｄに記憶させ、ＦＰＧＡデータメモリ４１ｄが記憶していた内容をＦＰＧＡデータメモリ４１ａに記憶させる、という動作をＦＰＧＡデータメモリ４１ａ～４１ｄに行わせるようにしてもよい。

【００５６】

また、ＦＰＧＡデータモジュール２－１～２－ｎは、ローダ３を介さず予めＦＰＧＡデータメモリに格納されることとして、ＦＰＧＡデータメモリが形成するルックアップテーブル自体をシフトジスタが循環シフトさせるようにしてもよい。

この場合、図４に示すように、この演算システムはローダ３を省略してもよく、従って制御部４７はローダ３を制御する必要はない。一方、シフトレジスタ４０は、ｎ個のＦＰＧＡデータモジュールを格納するためのｎ個のＦＰＧＡデータメモリ４１－１～４１－ｎを備え、ＦＰＧＡデータメモリ４１－１～４１－ｎを指定の対象として循環シフトさせるものとする。

ＦＰＧＡデータメモリ４１－１～４１－ｎの構成は、上述のＦＰＧＡデータメモリ４１ａ～４１ｄと実質的に同一であるものとする。ただし、ＦＰＧＡデータメモリ４１－１～４１－ｎは、必ずしも記憶内容を書き換え可能である必要はない。

なお、図４の構成においても、ＦＰＧＡデータメモリ４１－１～４１－ｎはシフトレジスタ４０とは別個のものとしてランダムアクセス可能なメモリより構成されていてもよい。この場合、この演算システムは、例えば、シフトレジスタ４０に代えて、図２に示すものと実質的に同一のセレクタ４２を備えていればよい。

【００５７】

また、図１～図４の構成では、ソースプログラム５－１～５－ｎをそれぞれコンパイルしたＦＰＧＡデータモジュール２－１～２－ｎを、ＦＰＧＡデバイス４のＦＰＧＡデータメモリ４１ａ～４１ｄに適宜ロードしていくものとしていた。これに対して、ソースプログラム５－１～５－ｎをそのままロードするようにした演算システムを構成することもできる。図５は、このような場合の演算システムの構成を示す。

【００５８】

この演算システムでは、ローダ３’は、制御部４７’からの指示に基づいて、プログラム記憶部５に記憶されたモジュール別のソースプログラム５－１～５－ｎを適宜メモリ４１ａ’～４１ｄ’にロードし、シフトレジスタ４０’は、制御部４７’からの指示に基づいて、メモリ４１ａ’～４１ｄ’のうち、実行すべきソースプログラムを記憶しているものを指定する。インタプリタ４８は、シフトレジスタ４０’に指定されたメモリにロードされたソースプログラム中の命令を１命令ずつ順次解釈し、その解釈結果に従ってゲートアレイ４３’を構成するゲ

ート回路４３ａに論理構成を行わせるべく所定の信号を出力する。解釈の結果、他のモジュールのソースプログラムを呼び出す命令であった場合には、その旨を制御部４７’に通知する。

【００５９】

制御部４７’は、他のモジュールの呼び出しが通知されると、ゲートアレイ４３’の内部状態（フリップフロップ４３ｂに保持されているデータ）と、呼び出し元のソースプログラムのモジュールを識別するためのデータと、次に実行をすべき命令を示すデータを退避スタック４５に退避すると共に、フリップフロップ４３ｂに保持されているデータのうち呼び出し先のモジュールに引数として渡すものを、引数受け渡し部４６に一時保存させる。そして、呼び出し先のソースプログラムが既にロードされているか否かを判別し、ロードされていなければ、ローダ３’に呼び出し先のソースプログラムをロードさせ、シフトレジスタ４０’に、呼び出し先のソースプログラムがロードされたメモリを指定させて、引数受け渡し部４６に一時保存されたデータを入力データとしてゲートアレイ４３’に与える。一方、ロードされていれば、呼び出し先のソースプログラムがロードされたメモリをシフトレジスタ４０’に指定させて、引数受け渡し部４６に一時保存されたデータを入力データとしてゲートアレイ４３’に与える。

【００６０】

また、呼び出し先のソースプログラムに従った演算が終了すると、ゲートアレイ４３’からの出力データを呼び出し元のモジュールに渡す引数として引数受け渡し部４６に一時保存させる。そして、退避スタック４５に退避されたデータに従って、呼び出し元のソースプログラムを記憶するメモリを再びシフトレジスタ４０’に指定させ、退避スタック４５に退避された内部状態をフリップフロップ４３ｂに戻し、引数受け渡し部４６に一時保存された引数をフリップフロップ４３ｂのうちの所定のものに書き込ませる。そして、退避スタック４５に退避されたデータに基づいて呼び出し元のモジュールのソースプログラムに従った演算を再開させる。

【００６１】

なお、インタプリタ４８は、複数のゲート回路の組み合わせによるハードウェ

アで構成することができ、その出力によってゲートアレイ４３’に含まれるゲート回路４３ａの論理構成を、演算の実行速度にほとんど影響を与えることなく高速に行うことができる。また、ここでのゲートアレイ４３’は、ソースプログラム中の各命令を終了したときのデータをフリップフロップ４３ｂの所定のものに保持させることで、各命令を順次実行していくことができる。

【００６２】

以上のようにインタプリタ４８を含む構成とすることによって、ソースプログラム５－１～５－ｎをモジュール別に順次ＦＰＧＡデバイス４’にロードしていくことが可能となる。このため、ＦＰＧＡデバイス４’の構成に合わせたコンパイラがなくても、複数のモジュールからなる大規模なプログラムに従った演算を、ハードウェア的に高速に行うことが可能となる。

さらに、図５の演算システムはメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、メモリが単一でＦＰＧＡデータモジュールの呼び出しがある毎に新たなソースプログラムがロードし直される構成に比べても演算が効率的で高速になる。

【００６３】

なお、図５の構成においても、メモリ４１ａ’～４１ｄ’はシフトレジスタ４０’とは別個のものとして、ランダムアクセス可能なメモリより構成されていてもよい。この場合、例えば図６に示すように、この演算システムは、シフトレジスタ４０’に代えて、図２の構成におけるセレクタ４２と実質的に同一の動作を行うセレクタ４２’を備えるものとしてもよい。

【００６４】

また、メモリ４１ａ’～４１ｄ’は図３に示すＦＰＧＡデータメモリ４１ａ～４１ｄと実質的に同一の構成を有していてもよく、この場合、シフトレジスタ４０’は、図３の構成におけるシフトレジスタ４０と実質的に同一の動作を行えばよい。

【００６５】

また、ソースプログラム５－１～５－ｎは、ローダ３’を介さず予めメモリに

格納することとしてもよい。

この場合、図７に示すように、この演算システムはローダ３’を省略できる。一方、シフトレジスタ４０’は、ｎ個のソースプログラムを格納するためのｎ個のメモリ４１－１’～４１－ｎ’を備え、これらｎ個のメモリを指定の対象として循環シフトさせるものとする。メモリ４１－１’～４１－ｎ’の構成は、上述のＦＰＧＡデータメモリ４１－１～４１－ｎと実質的に同一であるものとする。

【００６６】

なお、図７の構成においても、メモリ４１－１’～４１－ｎ’はシフトレジスタ４０’とは別個のものとしてランダムアクセス可能なメモリより構成されていてもよい。この場合、この演算システムは、例えば、シフトレジスタ４０’に代えて、図６に示すものと実質的に同一のセレクタ４２’を備えていればよい。

【００６７】

【発明の効果】

以上説明したように本発明によれば、複数のプログラムモジュールからなる大規模なプログラムであっても、各プログラムモジュールを適時メモリにロードしていく仕組みを有するので、該プログラムに応じた演算の実行をハードウェアで実現することが可能となる。また、コンフィギュレーションの書き換えによるオーバーヘッドの影響がなく演算が効率的となる。

【図面の簡単な説明】

【図１】

本発明の実施の形態にかかる演算システムの構成を示すブロック図である。

【図２】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図３】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図４】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【図５】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【図６】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【図７】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【符号の説明】

１　演算システム

２　ＦＰＧＡデータ記憶部

３　ローダ

４　ＦＰＧＡデバイス

６．コンパイラ

２－１～２－ｎ　ＦＰＧＡデータモジュール

４０　シフトレジスタ

４１ａ～４１ｄ、４１－１～４１－ｎ　ＦＰＧＡデータメモリ

４２　セレクタ

４３　ゲートアレイ

４３ａ　ゲート回路

４３ｂ　フリップフロップ

４４　呼び出し検出部

４５　退避スタック

４６　引数受け渡し部

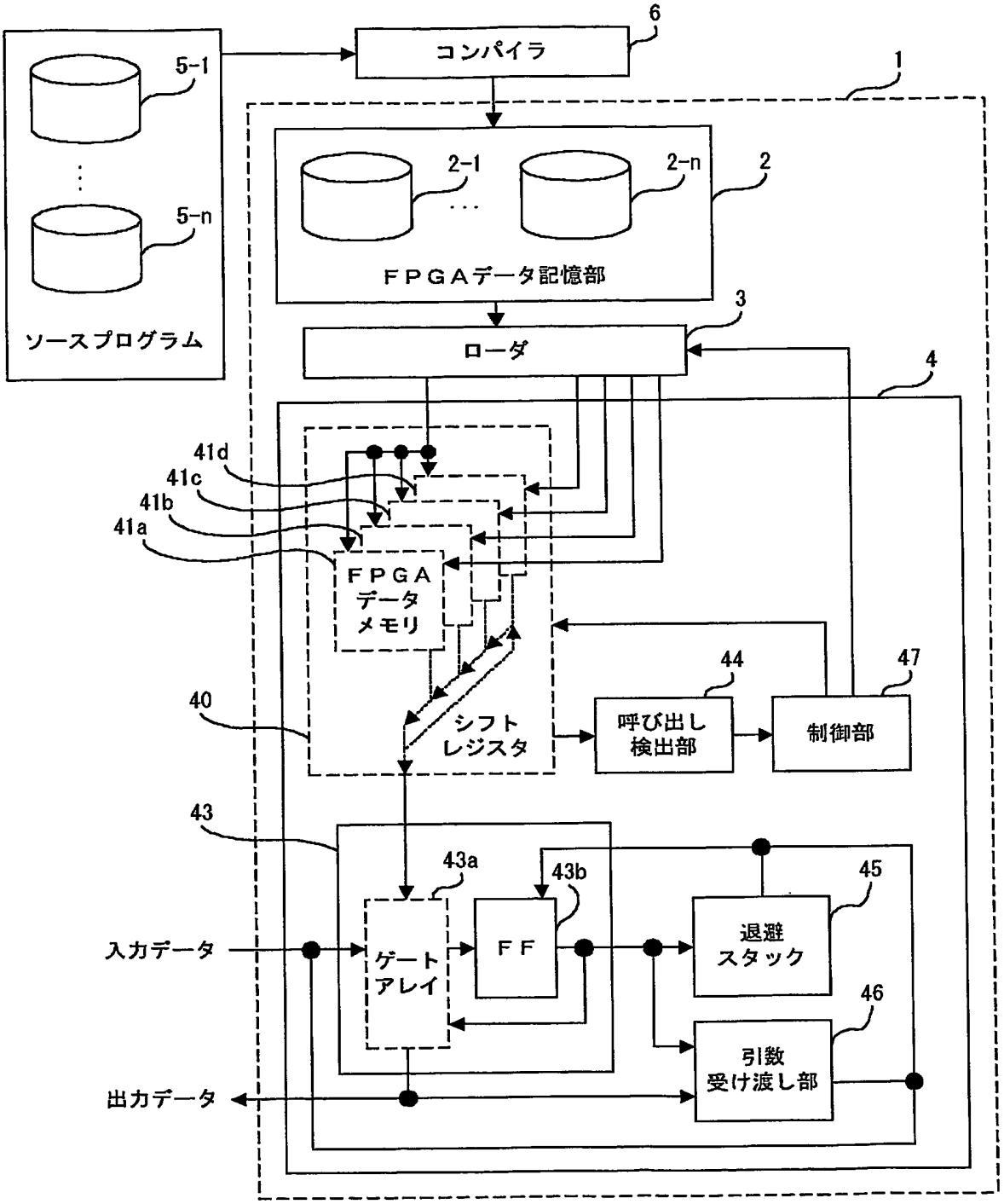
４７　制御部

４８　インタプリタ

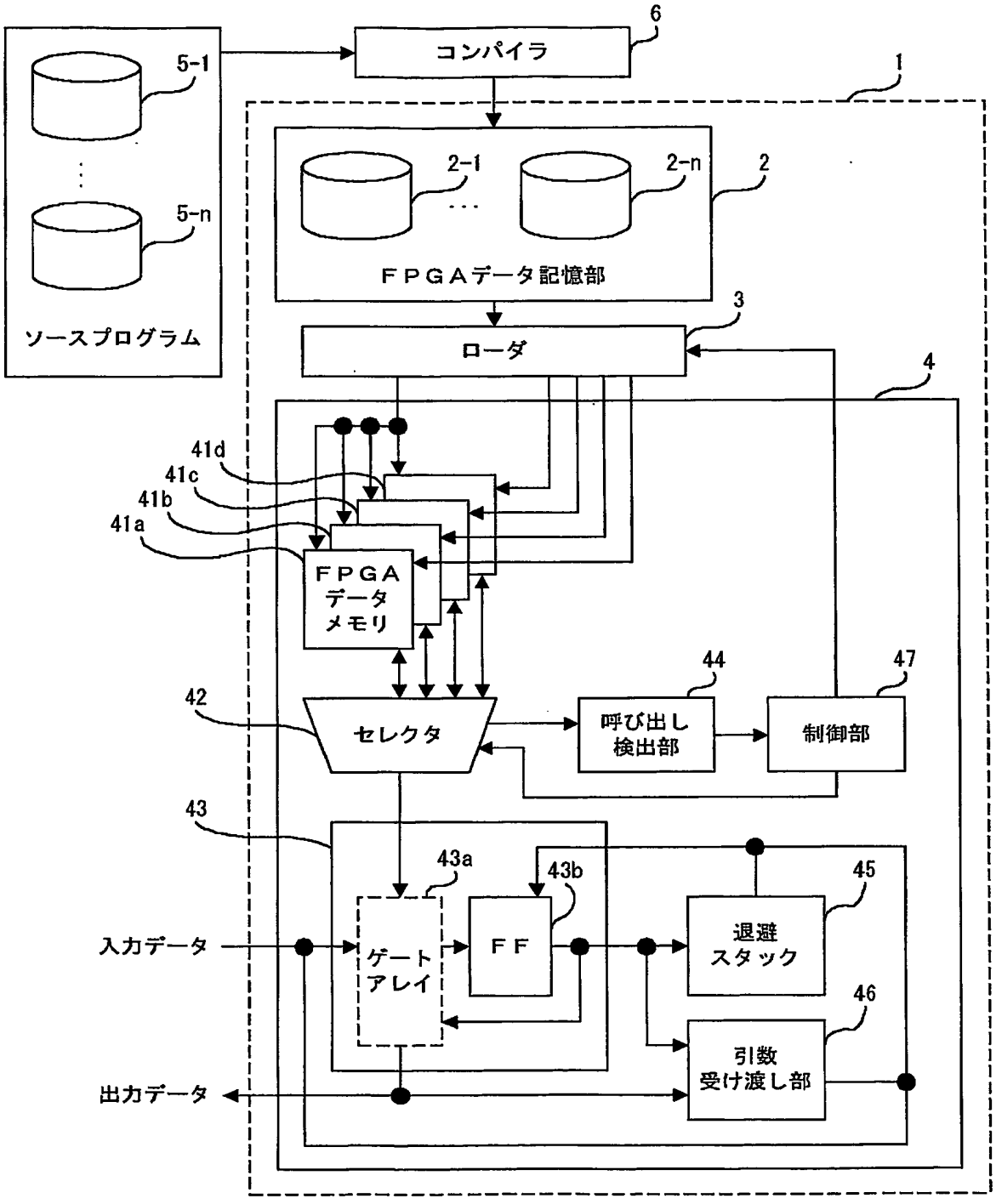
５－１～５－ｎ　ソースプログラム

【書類名】　図面

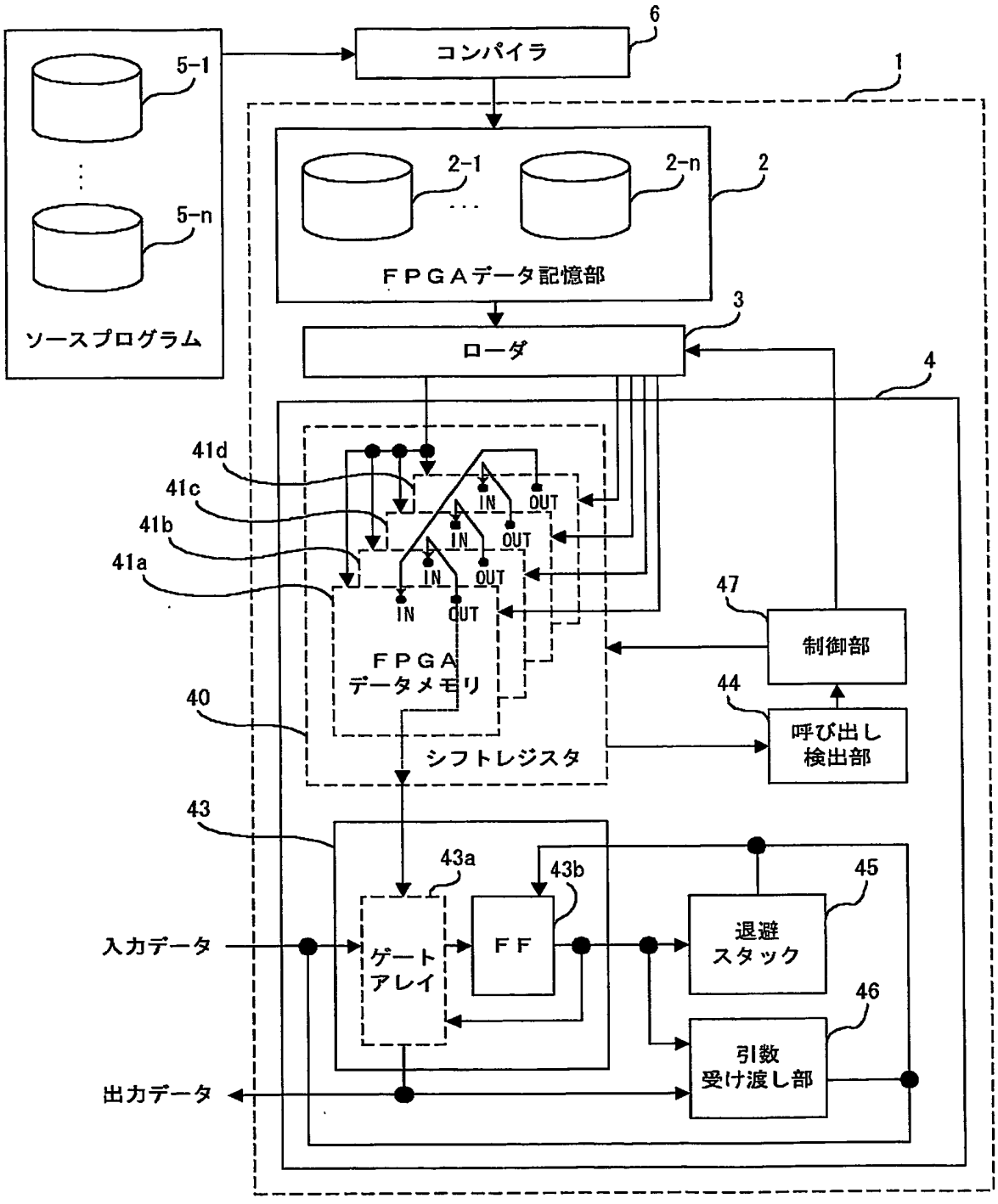
【図１】


6
コンパイラ
1
5-1
5-n
ソースプログラム
2-1
2-n
2
FPGAデータ記憶部
3
ローダ
4
41d
41c
41b
41a
FPGA
データ
メモリ
44
47
40
シフト
レジスタ
呼び出し
検出部
制御部
43
43a
43b
45
入力データ
ゲート
アレイ
FF
退避
スタック
46
引数
受け渡し部
出力データ

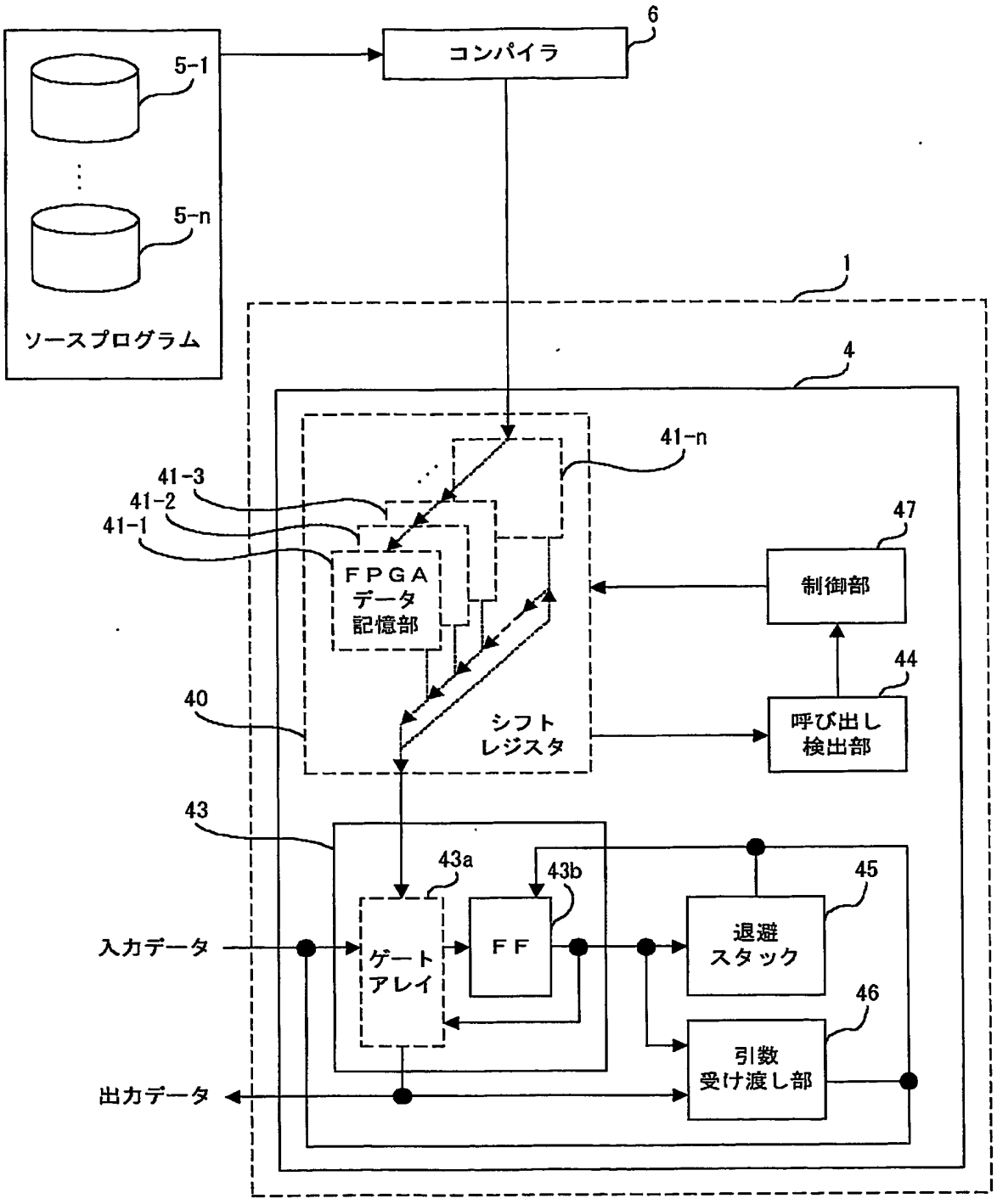
【図２】


6
コンパイラ
1
5-1
5-n
ソースプログラム
2-1
2-n
2
ＦＰＧＡデータ記憶部
3
ローダ
4
41d
41c
41b
41a
ＦＰＧＡ
データ
メモリ
44
47
42
セレクタ
呼び出し
検出部
制御部
43
43a
43b
45
入力データ
ゲート
アレイ
ＦＦ
退避
スタック
46
引数
受け渡し部
出力データ

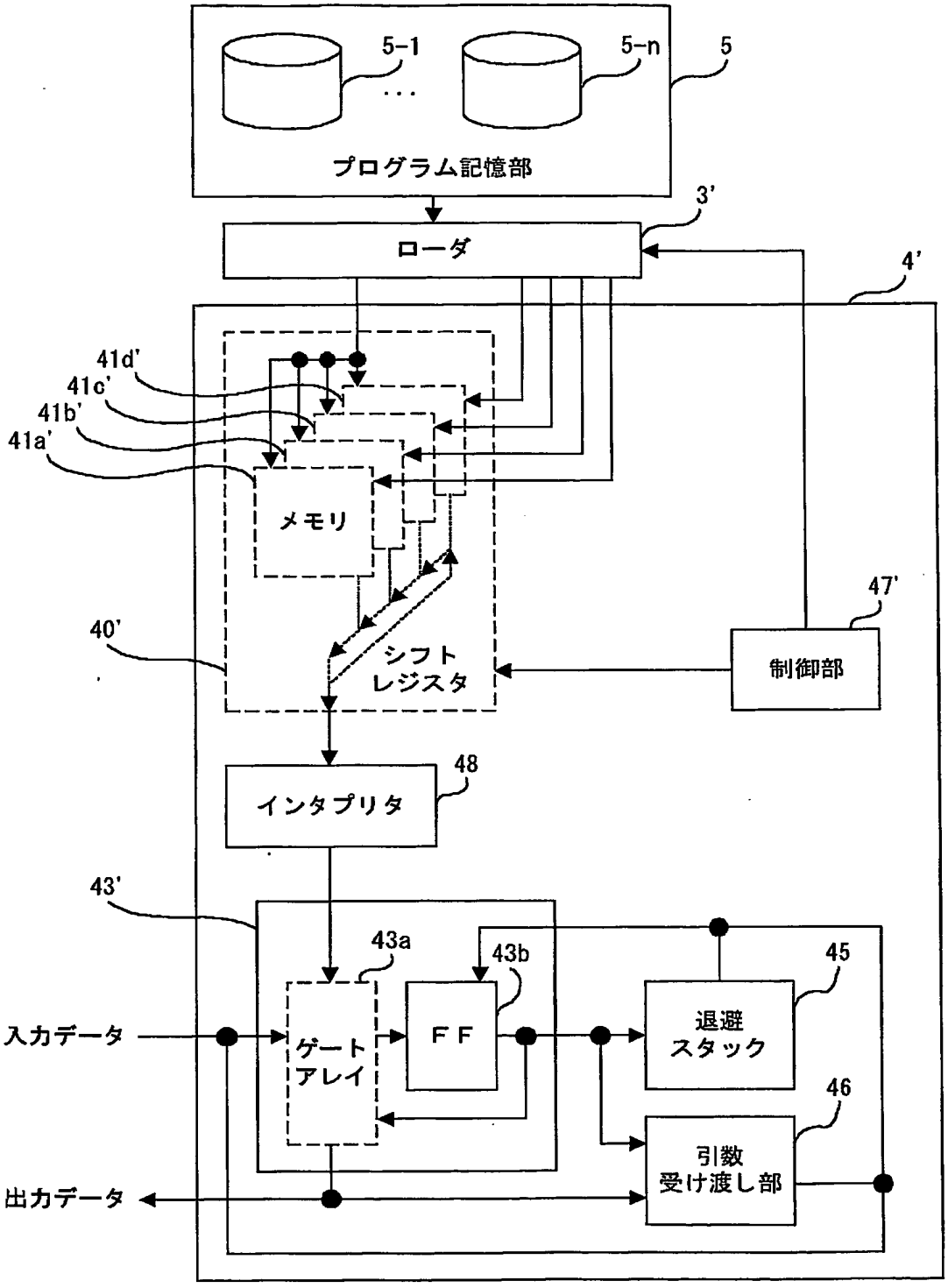
【図３】


6
コンパイラ
1
5-1
5-n
ソースプログラム
2-1
2-n
2
ＦＰＧＡデータ記憶部
3
ローダ
4
41d
41c
41b
41a
IN
OUT
ＦＰＧＡ
データメモリ
47
制御部
44
呼び出し
検出部
40
シフトレジスタ
43
43a
43b
45
入力データ
ゲート
アレイ
ＦＦ
退避
スタック
46
引数
受け渡し部
出力データ

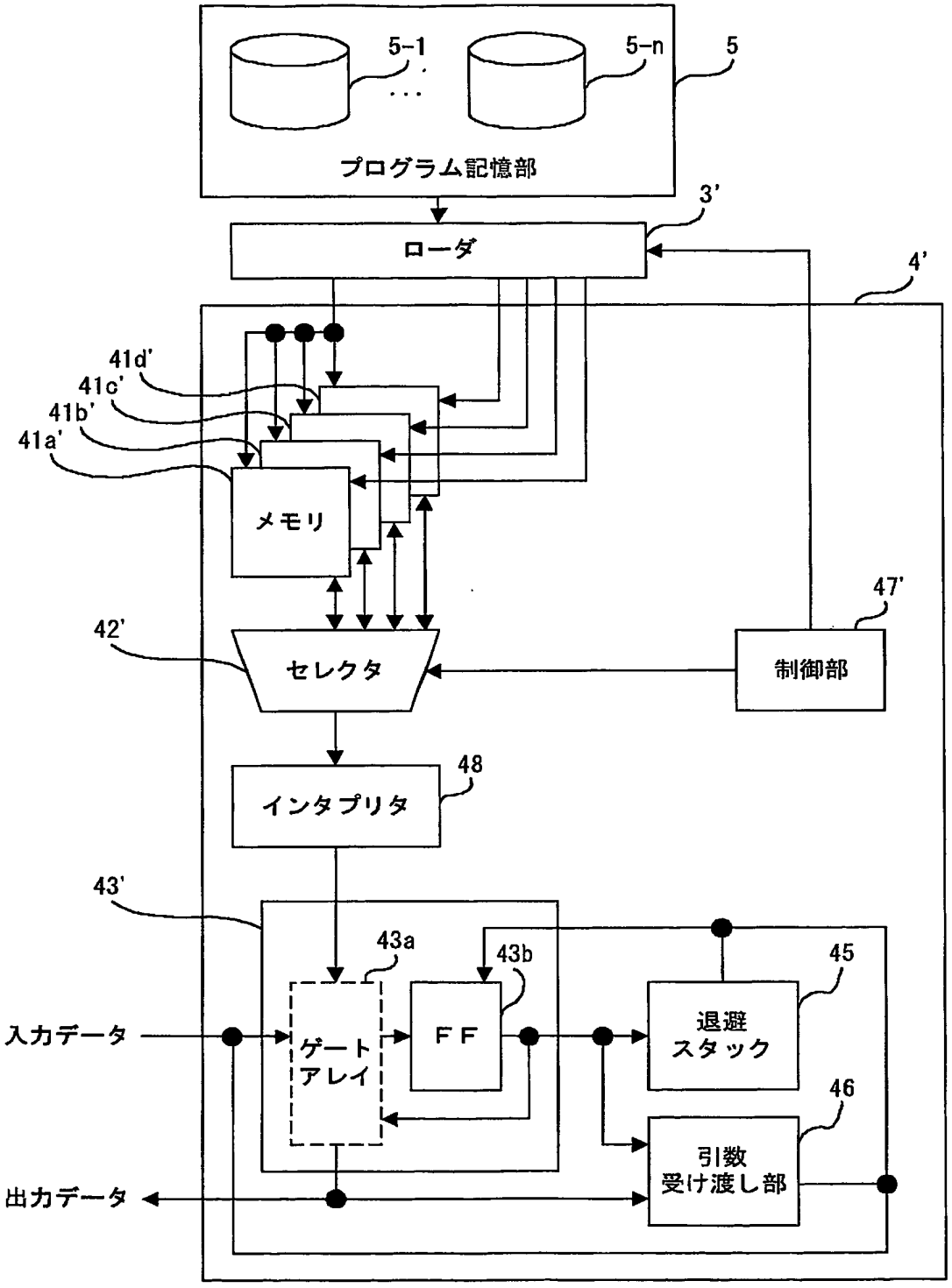
【図４】


6
コンパイラ
5-1
5-n
ソースプログラム
1
4
41-n
41-3
41-2
41-1
FPGA
データ
記憶部
47
制御部
44
呼び出し
検出部
40
シフト
レジスタ
43
43a
43b
45
入力データ
ゲート
アレイ
FF
退避
スタック
46
引数
受け渡し部
出力データ

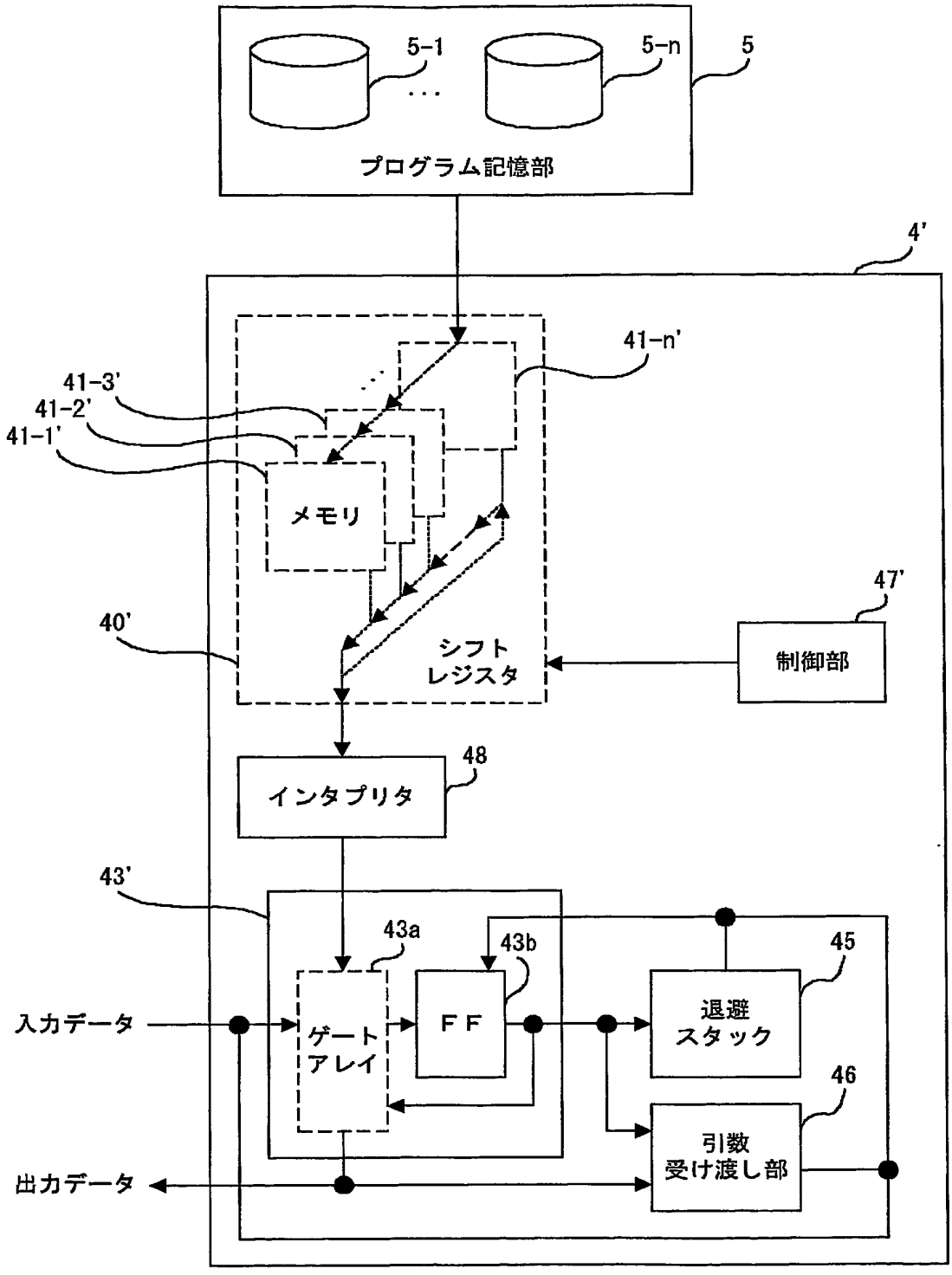
【図５】


5-1
5-n
5
プログラム記憶部
3'
ローダ
4'
41d'
41c'
41b'
41a'
メモリ
47'
40'
シフト
レジスタ
制御部
48
インタプリタ
43'
43a
43b
45
入力データ
ゲート
アレイ
ＦＦ
退避
スタック
46
引数
受け渡し部
出力データ

【図６】


5-1
5-n
5
プログラム記憶部
3'
ローダ
4'
41d'
41c'
41b'
41a'
メモリ
47'
42'
セレクタ
制御部
48
インタプリタ
43'
43a
43b
45
入力データ
ゲート
アレイ
ＦＦ
退避
スタック
46
引数
受け渡し部
出力データ

【図７】


5-1
5-n
5
プログラム記憶部
4'
41-n'
41-3'
41-2'
41-1'
メモリ
47'
40'
シフト
レジスタ
制御部
48
インタプリタ
43'
43a
43b
45
入力データ
ゲート
アレイ
ＦＦ
退避
スタック
46
引数
受け渡し部
出力データ

【書類名】　要約書

【要約】

【課題】　汎用のＣＰＵを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現する。

【解決手段】　ゲートアレイ４３は、シフトレジスタ４０に指定されたＦＰＧＡデータメモリに記憶されたＦＰＧＡデータモジュールに従ってハードウェア的に演算を行う。このＦＰＧＡデータメモリに記憶されたモジュールが他のモジュールを呼び出すものであることが呼び出し検出部４４によって検出されると、フリップフロップ４３ｂに保持された演算の途中結果のデータが退避スタック４５に退避され、呼び出し先のモジュールに渡す引数が引数受け渡し部４６に一時保存される。その後、呼び出し先のモジュールを記憶したＦＰＧＡデータメモリをシフトレジスタ４０に指定させ、呼び出し元のモジュールに復帰する際には、退避スタック４５に退避されたデータがフリップフロップ４３ｂに書き戻される。

【選択図】　図１

【書類名】　手続補正書
【提出日】　平成14年 1月24日
【あて先】　特許庁長官　殿
【事件の表示】
　【出願番号】　特願2001-401462
【補正をする者】
　【識別番号】　500323188
　【氏名又は名称】　東京エレクトロンデバイス株式会社
【補正をする者】
　【識別番号】　501186977
　【氏名又は名称】　西原　明法
【代理人】
　【識別番号】　100095407
　【弁理士】
　【氏名又は名称】　木村　満

【手続補正　1】

【補正対象書類名】　特許願

【補正対象項目名】　発明者

【補正方法】　変更

【補正の内容】

【発明者】

【住所又は居所】　神奈川県横浜市都筑区東方町１番地　東京エレクトロンデバイス株式会社内

【氏名】　三田　高司

【発明者】

【住所又は居所】　神奈川県川崎市中原区今井仲町３７４－４－５０５

【氏名】　西原　明法

【その他】　変更（追加）の理由は、代理人の不注意による記入漏れであります。

【プルーフの要否】　要

特願２００１－４０１４６２

出願人履歴情報

識別番号　［５００３２３１８８］

１．変更年月日　２０００年　７月　７日
［変更理由］　新規登録
住　所　神奈川県横浜市都筑区東方町１番地
氏　名　東京エレクトロンデバイス株式会社

特願２００１－４０１４６２

出　願　人　履　歴　情　報

識別番号　　［５０１１８６９７７］

１．変更年月日　　２００１年　５月１０日
［変更理由］　　新規登録
住　所　　神奈川県川崎市中原区今井仲町３７４－４－５０５
氏　名　　西原　明法